ELSEVIER

Theoretical Computer Science 265 (2001) 109–129

Theoretical Computer Science

www.elsevier.com/locate/tcs

Rigorous results for random $(2 + p)$-SAT

Dimitris Achlioptas[a, *, 1], Lefteris M. Kirousis[b], Evangelos Kranakis[c], Danny Krizanc[c]

[a] *Microsoft Research, One Microsoft Way, Redmond WA 98052, USA*
[b] *Department of Computer Engineering and Informatics, University of Patras, University Campus, GR-265 04 Patras, Greece*
[c] *School of Computer Science, Carleton University, Ottawa, Ontario, Canada K1S 5B6*

Abstract

In recent years there has been significant interest in the study of random k-SAT formulae. For a given set of n Boolean variables, let B_k denote the set of all possible disjunctions of k distinct, non-complementary literals from its variables (k-clauses). A random k-SAT formula $F_k(n, m)$ is formed by selecting uniformly and independently m clauses from B_k and taking their conjunction. Motivated by insights from statistical mechanics that suggest a possible relationship between the "order" of phase transitions and computational complexity, Monasson and Zecchina (Phys. Rev. E 56(2) (1997) 1357) proposed the random $(2+p)$-SAT model: for a given $p \in [0, 1]$, a random $(2 + p)$-SAT formula, $F_{2+p}(n, m)$, has m randomly chosen clauses over n variables, where pm clauses are chosen from B_3 and $(1 - p)m$ from B_2. Using the heuristic "replica method" of statistical mechanics, Monasson and Zecchina gave a number of non-rigorous predictions on the behavior of random $(2 + p)$-SAT formulae. In this paper we give the first *rigorous* results for random $(2 + p)$-SAT, including the following surprising fact: for $p \leqslant 2/5$, with probability $1 - o(1)$, a random $(2 + p)$-SAT formula is satisfiable iff its 2-SAT subformula is satisfiable. That is, for $p \leqslant 2/5$, random $(2 + p)$-SAT behaves like random 2-SAT.

1. Introduction

The problem of determining the satisfiability of Boolean formulae is central to the understanding of computational complexity. Moreover, it is of tremendous practical interest as it arises naturally in numerous settings. Typically, formulae are considered to be in conjunctive normal form (CNF), i.e. a conjunction of disjunctions (clauses), and one needs to determine if there exists an assignment of truth values to the formula's

* Corresponding author.
E-mail addresses: optas@microsoft.com (D. Achlioptas), kirousis@ceid.upatras.gr (L.M. Kirousis), kranakis@scs.carleton.ca (E. Kranakis), krizanc@scs.carleton.ca (D. Krizanc).

[1] Research performed while at the Department of Computer Science, University of Toronto.

0304-3975/01/$ - see front matter
PII: S0304-3975(01)00154-2

variables so that at least one literal is satisfied from each clause. Cook's Theorem [12] asserts that satisfiability is NP-complete and thus "as hard" as any problem whose solutions can be verified in polynomial time. A canonical version of the satisfiability problem is k-SAT, where each clause of the input formula has precisely k literals. Cook proved that for $k \geqslant 3$, k-SAT is NP-complete, while for $k = 2$ it can be solved in polynomial time [12].

Given that satisfiability is NP-complete, practitioners seek heuristic solutions to the problem of deciding the satisfiability of large formulae. The most common approach is to employ some variation of the Davis–Putnam (DP) algorithm [15, 14]. In the last two decades, partly in order to evaluate and improve satisfiability algorithms, there has been considerable work in analyzing the satisfiability of random formulae. In fact, some early results suggested that deciding satisfiability is "easy on average". Unfortunately, while "easy" is easy to interpret, "on average" is not.

One of the earliest and most often quoted results for satisfiability being easy on average is due to Goldberg [25]. In [21], though, Franco and Paull pointed out that the distribution of instances used in the analysis of [25] is so greatly dominated by "very satisfiable" formulae that if one tries truth assignments completely at random, the expected number of trials until finding a satisfying one is O(1). Moreover, in [21] the performance of the DP algorithm on random instances of k-SAT was considered. In particular, for a given set of n Boolean variables, let B_k denote the set of all $2^k\binom{n}{k}$ possible disjunctions of k distinct, non-complementary literals on its variables (k-clauses). A random k-SAT formula $F_k(n,m)$ is formed by selecting uniformly, independently, and with replacement[2] m clauses from B_k and taking their conjunction. Franco and Paull [21] showed that for all $k \geqslant 3$ and every constant $r > 0$, with probability $1 - o(1)$, the DP algorithm takes an *exponential* number of steps to report the satisfying truth assignments of $F_k(n, rn)$, i.e. either to report all ("cylinders" of) solutions, or that no solutions exist.

In [36], Selman et al. gave extensive experimental evidence suggesting that for $k \geqslant 3$, there is a range of the clauses-to-variables ratio, r, within which it seems hard even to *decide* if a randomly chosen k-SAT instance is satisfiable or not (as opposed to finding all satisfying truth assignments). For example, for $k = 3$ their experiments draw the following remarkable picture: for $r < 4$, a satisfying truth assignment can be easily found for almost all formulae; for $r > 4.5$, almost all formulae are unsatisfiable; for $r \approx 4.2$, a satisfying truth assignment can be found for roughly half the formulae and around this point the computational effort for finding a satisfying truth assignment, whenever one exists, is maximized.

We will be interested in random formulae from an asymptotic point of view, i.e. as the number of variables grows. In particular, we will say that a sequence of random events $\mathscr{E}_n$ occurs almost surely (a.s.) if $\lim \Pr[\mathscr{E}_n] = 1$. If $\liminf_{n\to\infty} \Pr[\mathscr{E}_n] > 0$, we will

[2] Allowing replacement simplifies calculations greatly. Moreover, in the interesting range $m = \Theta(n)$ the expected number of repeated clauses is O(1) and thus it is virtually inconsequential. In particular, all the results discussed in this paper hold also in the setting where replacement is not allowed.

say that $\mathscr{E}_n$ occurs with positive probability. Let $g_k(n,r)$ denote the probability that $F_k(n,rn)$ is satisfiable. In [11], the following compelling possibility was put forward and by now has become a folklore conjecture.

Conjecture 1 (*Satisfiability threshold conjecture*). For every $k \geqslant 2$, there exists a constant r_k such that for any $\varepsilon > 0$,

$$\lim_{n\to\infty} g_k(n, r_k - \varepsilon) = 1 \quad \text{and} \quad \lim_{n\to\infty} g_k(n, r_k + \varepsilon) = 0.$$

The satisfiability threshold conjecture, which motivates our work, has attracted a lot of attention in computer science, mathematics and, more recently, in mathematical physics [31–34]. For the connections of random formulae to proof-complexity and computational-hardness we refer the interested reader to the excellent surveys by Beame and Pitassi [5] and Cook and Mitchell [13], respectively.

The rest of the paper is organized as follows. In Section 2 we summarize most known rigorous results regarding the conjecture. In Section 3 we first discuss how insights on the conjecture derived by using techniques and notions from statistical mechanics have motivated the random $(2+p)$-SAT model. Then we describe our contributions and their relationship to the non-rigorous results on random $(2+p)$-SAT. Finally, in Sections 4 and 5 we prove our results, by giving conditions for almost sure unsatisfiability and almost sure satisfiability of $F_{2+p}(n,m)$, respectively.

2. Summary of known results for random k-SAT

2.1. Random 2-SAT

For $k=2$, Chvátal and Reed [11], Goerdt [24] and Fernandez de la Vega [19] independently proved Conjecture 1, in fact determining $r_2 = 1$. Note that 2-SAT being solvable in polynomial time is equivalent to saying that for $k=2$ we have a simple characterization of unsatisfiable 2-SAT formulae [12]. This fact enables a direct/combinatorial attack of random 2-SAT that focuses on the emergence of the "most likely" unsatisfiable subformulae in the evolution of $F_2(n,rn)$. Indeed, this was the approach in [11, 24]. More recently, Bollobás et al. [7] also used this approach to determine the "scaling window" for random 2-SAT: the transition from almost sure satisfiability to almost sure unsatisfiability occurs at $m = n + \lambda n^{2/3}$ as λ goes from $-\infty$ to $+\infty$.

2.2. Random 3-SAT

For $k \geqslant 3$, much less progress has been made towards Conjecture 1. Neither the value, nor even the existence of r_k has been established. In the following, by $r_k > r^*$ we will mean that for $r \leqslant r^*$, $F_k(n,rn)$ is a.s. satisfiable (analogously for $r_k < r^*$).

The first upper bound for r_3 was given by Franco and Paull [21], who observed that the expected number of satisfying truth assignments of $F_3(n,rn)$, $(2(7/8)^r)^n$, is

o(1) when $r > r^0 = 5.191\ldots$. In [8], Broder et al. showed $r_3 < r^0 - 10^{-7}$, i.e. that the expectation bound is not tight. Shortly afterwards, El-Maftouhi and Fernandez de la Vega [18] proved $r_3 < 5.08$ and, independently, Kamath et al. [27] proved $r_3 < 4.758$. This was further improved to $r_3 < 4.601$ by Kirousis et al. [29], using a much more direct and simple approach than [18, 27]. We will apply the method of [29] in Section 4 and elaborate on it, therein. Independently, Dubois and Boufkhad [16] obtained $r_3 < 4.64$ with a method similar to that of Kirousis et al. By estimating exactly a hypergeometric sum appearing in [29], Janson et al. [26] proved $r_3 < 4.598$. Zito [38] improved the bound to less than 4.579 by combining the approaches of [16] and [29]. Finally, by combining the approach of [29] with tight bounds for the occupancy problem [27], Kaporis et al. [20] gave the best known bound, $r_3 < 4.571$. Recently, Dubois et al. [17] announced $r_3 < 4.506$.

Unlike upper bounds, which come from probabilistic counting arguments, all known lower bounds for r_3 are algorithmic. The first analysis of an algorithm on $F_3(n, rn)$ was given by Chao and Franco [9] who showed that the UNIT CLAUSE (UC) algorithm has positive probability of finding a satisfying truth assignment for $r < 8/3$ and, when combined with a "majority" rule, for $r < 2.9$. Note that since UC succeeds with positive probability instead of a.s. this did not imply $r_3 \geqslant 2.9$; their analysis, though, inspired a number of subsequent papers [10, 11, 23, 3, 4, 1].

The first lower bound for r_3 was given by Franco [20] who considered the "pure literal" heuristic on $F_3(n, rn)$. This heuristic satisfies a literal only if its complement does not appear in the formula, thus only making "safe" steps. He showed that for $r < 1$, the pure literal heuristic eventually sets all the variables, yielding $r_3 \geqslant 1$. Broder et al. [8] proved $r_3 \geqslant 1.63$, by showing that the pure literal heuristic a.s. sets all the variables for $r < 1.63$. Later, Frieze and Suen [23] analyzed a generalization of UC, called GUC, and gave an exact analysis of its probability of success. They showed that for $r < 3.003$, GUC succeeds with positive probability. Moreover, they proved that a modified version of GUC, performing a very limited form of backtracking, succeeds a.s. for such r, thus yielding $r_3 > 3.003$. Recently, the first author [1] introduced a new heuristic for 3-SAT that sets two variables at a time and by analyzing its performance on $F_3(n, rn)$ proved $r_3 > 3.145$, the best known lower bound for random 3-SAT.

A big step towards proving the existence of r_k was made by Friedgut [22]. Recall that $g_k(n, r)$ is the probability that $F_k(n, rn)$ is satisfiable. [3]

Theorem 1 (Friedgut [22]). *For every $k \geqslant 2$, there exists $r_k(n)$ such that for any $\varepsilon > 0$,*

$$\lim_{n\to\infty} g_k(n, r_k(n) - \varepsilon) = 1 \quad \textit{and} \quad \lim_{n\to\infty} g_k(n, r_k(n) + \varepsilon) = 0.$$

[3] Theorem 1 was proven for the model where each of the $N_k = 2^k \binom{n}{k}$ clauses appears independently of all others in the random formula with probability rn/N_k. As we will see in Section 5, the sharp threshold for that model easily transfers to $F_k(n, rn)$.

The following immediate corollary of Theorem 1 is very useful in bounding r_k from below. It implies that if for some r^*, $F_k(n, r^*n)$ is satisfiable with positive probability then $r_k \geqslant r^*$.

Corollary 1. *If* $\lim_{n\to\infty} \inf g_k(n,r) > 0$ *then for any* $\varepsilon > 0$, $\lim_{n\to\infty} g_k(n, r-\varepsilon) = 1$.

Note that combining Corollary 1 with the results in [9] we get $r_3 \geqslant 8/3$ and $r_3 \geqslant 2.9$, for each respective algorithm. By now, though, these bounds have been superseded by the results in [23, 1].

3. The $(2+p)$-SAT model

3.1. The replica method and motivation

All of our previous discussion pertains to mathematical (rigorous) results. If one is willing to settle for non-rigorous results, then substantially more can be said. In particular, all the results we discuss in this section are based on the non-rigorous "replica method" of statistical mechanics. While the replica method is a sophisticated mathematical methodology, its validity rests on a number of *unproven* assumptions; moreover, these assumptions are known to be false in general, and there is very little understanding of when they might be valid. Besides this fundamental objection, in applying the replica method it is very often necessary to make numerical approximations of intermediate results, without being able to provide non-trivial bounds on their accuracy.

In [31, 32], Monasson and Zecchina showed how to apply the replica method to the random k-SAT problem. By relating the "energy" of a truth assignment to the number of clauses it fails to satisfy, they get $r_2 = 1$ and give improved upper and lower bounds for r_k for small $k \geqslant 3$. Given a formula F and a variable x, let us say that x is *frozen* in F if all truth assignments that satisfy as many of the clauses of F as possible, assign x the same value. Treating the fraction of frozen variables as an "order parameter", Monasson and Zecchina [32] claimed that the "phase transition" from almost sure satisfiability into almost sure unsatisfiability is of "second order" for $k = 2$, but of "first order" for $k \geqslant 3$. In mathematical terms this amounts to the following. For $r > r_k$, $F_k(n, rn)$ a.s. has $f_k(r)\cdot n + o(n)$ frozen variables. Moreover, $\lim_{r\to r_2^+} f_2(r) = 0$, but for $k \geqslant 3$, $\lim_{r\to r_k^+} f_k(r) > 0$. That is, above the threshold, the fraction of frozen variables "takes off" in a continuous manner for $k = 2$, but in a discontinuous one for $k \geqslant 3$.

In [32], as an attempt to illuminate the difference in the order of the phase transition for $k = 2$ vs. $k = 3$, the $(2+p)$-SAT model was introduced: fix $p \in [0,1]$; similarly to random k-SAT we have a random formula, $F_{2+p}(n,m)$, with n variables and m clauses chosen uniformly and independently with replacement, but now pm clauses are chosen from the set of all clauses of length 3 (B_3) and $(1-p)m$ from the set of all clauses of length 2 (B_2). Thus, $p = 0$ corresponds to random 2-SAT, while $p = 1$ corresponds

to random 3-SAT. Using the replica method, Monasson et al. [34] claimed that for *every* $p \in [0,1]$ there is a critical value of r, denoted by r_p, around which $F_{2+p}(n, rn)$ undergoes a phase transition, turning from a.s. satisfiable into a.s. unsatisfiable.

An easy upper bound on r_p follows from the fact that for $F_{2+p}(n, rn)$ to be satisfiable, both its 2-SAT and its 3-SAT subformula must be satisfiable independently of one another. In particular, since $F_{2+p}(n, rn)$ contains $r(1-p)n$ 2-clauses, just considering the 2-SAT subformula implies that $r_p(1-p) \leqslant r_2$, i.e. $r_p \leqslant 1/(1-p)$. Most remarkably, in [34], it was claimed that there exists $p^* > 0$, such that for all $p \in [0, p^*)$,

$$r_p = \frac{1}{1-p}.$$

In other words, for $p < p^*$ the rpn 3-clauses in $F_{2+p}(n, rn)$ are a.s. "irrelevant" to the formula's satisfiability.

To make this more precise, let us say that for a given p random $(2+p)$-SAT "behaves like random 2-SAT" if $F_{2+p}(n, m)$ is a.s. satisfiable if and only if $r < 1/(1-p)$. Let

$$p_c = \sup\{p\colon F_{2+p}(n, rn) \text{ is a.s. satisfiable iff } r < 1/(1-p)\}.$$

In [34] it was further claimed that $p_c = 0.413\ldots$. If true, this would imply that for *every* $\varepsilon > 0$, we can add $0.703n$ random 3-clauses to $F_2(n, (1-\varepsilon)n)$ and still have an a.s. satisfiable formula!

3.2. Our results

We first prove that for all $p \in [0,1]$, random $(2+p)$-SAT exhibits a sharp threshold. For $p \in [0,1]$ let $g_p(n, r)$ denote the probability that $F_{2+p}(n, rn)$ is satisfiable.

Theorem 2. *For every* $p \in [0,1]$, *there exists* $r_p(n)$ *such that for any* $\varepsilon > 0$,

$$\lim_{n\to\infty} g_p(n, r_p(n) - \varepsilon) = 1 \quad \textit{and} \quad \lim_{n\to\infty} g_p(n, r_p(n) + \varepsilon) = 0.$$

Much more surprisingly, we prove that $p_c \geqslant 2/5$. As a result, we establish that for all $\varepsilon > 0$, one can indeed add $(2/3)n$ random 3-clauses to $F_2(n, (1-\varepsilon)n)$ and still have an a.s. satisfiable formula.

Theorem 3. *For* $p \leqslant 2/5$,

$$r_p = \frac{1}{1-p}.$$

In [34], it was claimed that $0.413\ldots$ separates a "2-SAT-like" behavior from a "3-SAT-like" one. More precisely, it is claimed that for $p < 0.413\ldots$ the phase transition from almost sure satisfiability to almost sure unsatisfiability is of second order (like random 2-SAT), but for $p > 0.413\ldots$ it is of first order (like random 3-SAT). Interpreting "behaves like random 2-SAT" as "$F_{2+p}(n, rn)$ is a.s. satisfiable if and only

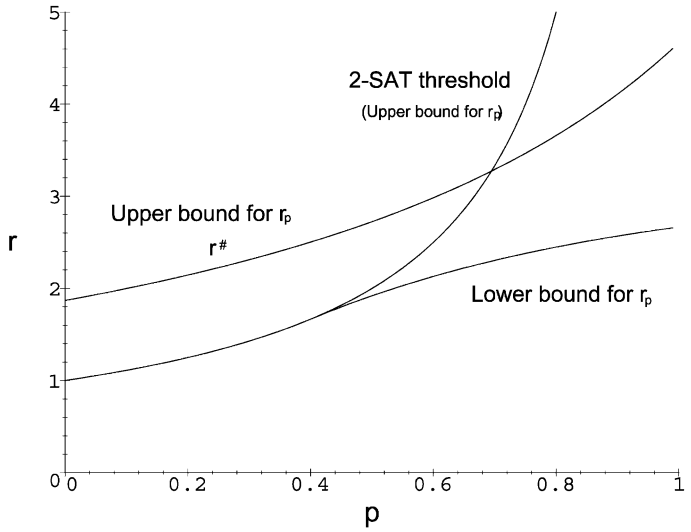

Fig. 1. Upper and lower bounds for r_p.

if $r<1/(1-p)$", we prove that while $(2+p)$-SAT behaves like 2-SAT for $p \leqslant 2/5$, this is not the case for $p \geqslant 0.694\ldots$. More generally, for $p>2/5$ we provide upper and lower bounds for r_p, the former implying $p_c<0.695$.

Theorem 4. *For* $p>2/5$,

$$\frac{24p}{(p+2)^2} \leqslant r_p \leqslant \min\left(\frac{1}{1-p}, r^{\#}\right),$$

where $r^{\#}$ *is the solution of* $(7/6)^{rp}(3/4)^r(2-\mathrm{e}^{-r(2/3-5p/21)})=1$.

These are the first rigorous results for random $(2+p)$-SAT. Theorems 3 and 4 are illustrated in Fig. 1.

Notation. Before we proceed to prove Theorems 2–4 we need to introduce some notation. For a literal l, var(l) will denote its underlying variable. For a set V of n Boolean variables, $B_i(V)$ will denote the set of all $2^i\binom{n}{i}$ i-clauses on the variables of V. Unless otherwise stated, we consider $V=\{x_1,\ldots,x_n\}$ and write B_i for $B_i(V)$. Also, we let Bin(N,s) denote the Binomial random variable with N trials each having probability of success s.

4. Almost sure unsatisfiability

Since a random instance of $(2+p)$-SAT is unsatisfiable if either its 2-SAT or its 3-SAT subformula is unsatisfiable, the fact $r_2=1$ and the bound $r_3<4.506$

imply

$$r_p \leqslant \min(1/(1-p), 4.506/p) = \begin{cases} 1/(1-p) & \text{if } p < 0.818\ldots, \\ 4.506/p & \text{if } p > 0.818\ldots . \end{cases} \quad (1)$$

Thus, for $p > 0.818\ldots$, random $(2+p)$-SAT does not behave like random 2-SAT. In the following, we will lower this upper bound for p_c from $0.818\ldots$ to 0.695. In particular, our bound implies that there is a constant $\lambda < 1$, such that a random formula with λn 2-clauses and $2.28n$ 3-clauses is a.s. unsatisfiable.

Let $F \in F_{2+p}(n, rn)$ be a randomly chosen instance of $(2+p)$-SAT and let $\mathscr{A}_n$ denote the set of all possible 2^n truth assignments. Let $\mathscr{S}_n \subseteq \mathscr{A}_n$ denote the random set of satisfying truth assignments (solutions) for F. Since a fixed truth assignment A satisfies a randomly chosen k-clause with probability $1 - 2^{-k}$,

$$\Pr[A \in \mathscr{S}_n] = (7/8)^{rpn}(3/4)^{r(1-p)n}. \quad (2)$$

Thus,

$$\mathbf{E}[|\mathscr{S}_n|] = 2^n(7/8)^{rpn}(3/4)^{r(1-p)n} \equiv s(r, p)^n.$$

Since, by definition, $\Pr[|\mathscr{S}_n| > 0] \leqslant \mathbf{E}[|\mathscr{S}_n|]$, $F_{2+p}(n, rn)$ is a.s. unsatisfiable if $s(r, p) < 1$, i.e. if

$$r > \frac{\ln 2}{\ln 4/3 - p \ln 7/6}.$$

The right-hand side above is strictly less than $1/(1-p)$ for $p > 0.752\ldots$, already improving upon the upper bound for p_c given by (1).

The price paid for bounding $\Pr[|\mathscr{S}_n| > 0]$ by $\mathbf{E}[|\mathscr{S}_n|]$ is that formulae with a very large number of solutions, although they may occur with very small probability for $r > r_p$, contribute substantially to $E[|\mathscr{S}_n|]$. Clearly, if we could replace $\mathscr{S}_n$ by a "smaller" set, we would get a quantity closer to $\Pr[|\mathscr{S}_n| > 0]$ and hence a tighter bound. The technique of [29] does precisely that by counting only those $A \in \mathscr{S}_n$ that satisfy a certain "local maximality" condition.

Let a solution $A \in \mathscr{S}_n$ be called "locally maximum" if every truth assignment obtained from A by changing the value of exactly one variable from 0 to 1 is not a solution of F. Let $\mathscr{S}_n^\sharp \subseteq \mathscr{S}_n$ be the set of all locally maximum solutions of F. Note now, that if F is satisfiable then $S_n^\sharp \neq \emptyset$, since the lexicographically greatest truth assignment of F is in $S_n^\sharp$ by definition. Thus,

$$\Pr[F \text{ is satisfiable}] \leqslant \mathbf{E}[|\mathscr{S}_n^\sharp|] = \Pr[A \in \mathscr{S}_n] \sum_{A \in \mathscr{A}_n} \Pr[A \in \mathscr{S}_n^\sharp \,|\, A \in \mathscr{S}_n]. \quad (3)$$

For a truth assignment A that assigns value 0 to a variable v, let $A(v)$ denote the truth assignment obtained from A by changing the value of v from 0 to 1. Let us fix a truth assignment $A \in \mathscr{S}_n$ and try to change the value of a variable v from 0 to 1 in A. First, note that the fact $A \in \mathscr{S}_n$ excludes $\binom{n}{3}$ clauses of B_3 and $\binom{n}{2}$ clauses of B_2

from the conjuncts of F (those dissatisfied by A). Second, note that the event $A(v) \notin \mathscr{S}_n$ occurs iff among the rn clauses in F there is a clause not satisfied by $A(v)$. Such a clause must contain $\bar{v}$ and its remaining literals must be dissatisfied by A (since it was satisfied by A but not $A(v)$). Hence, there are $\binom{n-1}{2}$ such 3-clauses and $\binom{n-1}{1}$ 2-clauses implying

$$\Pr[A(v) \notin \mathscr{S}_n \mid A \in \mathscr{S}_n] = 1 - \left(1 - \frac{\binom{n-1}{2}}{7\binom{n}{3}}\right)^{rpn} \left(1 - \frac{\binom{n-1}{1}}{3\binom{n}{2}}\right)^{r(1-p)n}$$

$$= 1 - \left(1 - \frac{3}{7n}\right)^{rpn} \left(1 - \frac{2}{3n}\right)^{r(1-p)n}$$

$$= 1 - \mathrm{e}^{-(3rp/7 + 2r(1-p)/3)} + \mathrm{O}(n^{-1})$$

$$= 1 - \mathrm{e}^{-r(2/3 - 5p/21)} + \mathrm{O}(n^{-1}). \tag{4}$$

To bound $\Pr[A \in \mathscr{S}_n^{\sharp} \mid A \in \mathscr{S}_n]$ using (4), we need to bound the probability that for every variable v assigned 0 by A, $A(v) \notin \mathscr{S}_n$. Letting $z(A)$ denote the number of variables assigned 0 by A, we claim

$$\Pr[A \in \mathscr{S}_n^{\sharp} \mid A \in \mathscr{S}_n] \leqslant (1 - \mathrm{e}^{-r(2/3 + 5p/21)} + \mathrm{O}(n^{-1}))^{z(A)}. \tag{5}$$

To see this intuitively, first observe that the sets of clauses "blocking" each $A(v)$ from being in $\mathscr{S}_n$ are disjoint for distinct variables v. Now since the total number of clauses is fixed and the blocking of each 0 "consumes" at least one clause, the blocking events should be negatively correlated. To derive this formally we apply the following Theorem of McDiarmid [30].

Theorem 5 (McDiarmid [30]). *Let U, I be finite non-empty sets. Let $(X_u)_{u \in U}$ be a family of independent random variables, each taking values in some set containing I; and for each $i \in I$ let $S_i = \{u \in U \mid X_u = i\}$. Let $(\mathscr{F}_i)_{i \in I}$ be a family of collections of subsets of U such that each collection is either monotone increasing or monotone decreasing. Then*

$$\Pr\left[\bigwedge_{i \in I} S_i \in \mathscr{F}_i\right] \leqslant \prod_{i \in I} \Pr[S_i \in \mathscr{F}_i]. \tag{6}$$

In our application the m identical, independent experiments are the m choices of clauses that form F. Fixing an (arbitrary) enumeration $x_1, \ldots, x_{z(A)}$ of the variables assigned 0 by A, (5) follows by applying Theorem 5 with $U = \{1, \ldots, m\}$, (ii) $I = \{1, \ldots, z(A)\}$, (iii) $X_u = i$, if the uth clause of F is satisfied by A but not by $A(v_i)$ and 0 otherwise, and (iv) $\mathscr{F}_i = 2^U - \emptyset$, for all i.

Combining (2), (3) and (5) we now have

$$\begin{aligned}\Pr[F \text{ is satisfiable}] &\leqslant \Pr[A \in \mathscr{S}_n] \sum_{k=0}^{n} \binom{n}{k} \Pr[A \in \mathscr{S}_n^{\sharp} \mid A \in \mathscr{S}_n \text{ and } z(A) = k] \\ &\leqslant \Pr[A \in \mathscr{S}_n] \sum_{k=0}^{n} \binom{n}{k} (1 - \mathrm{e}^{-r(2/3-5p/21)} + \mathrm{O}(n^{-1}))^k \\ &= (7/8)^{rpn}(3/4)^{r(1-p)n}(2 - \mathrm{e}^{-r(2/3-5p/21)} + \mathrm{O}(n^{-1}))^n \\ &= \mathrm{O}(d(r,p)^n), \end{aligned} \tag{7}$$

where $d(r,p) = (7/6)^{rp}(3/4)^r(2 - \mathrm{e}^{-r(2/3-5p/21)})$. Therefore, $F_{2+p}(n, rn)$ is a.s. unsatisfiable if $d(r,p) < 1$. Letting $r^{\#}$ denote the solution of $d(r,p) = 1$ as a function of p, and taking into account the trivial bound $r_p \leqslant 1/(1-p)$, we get the upper bound in Theorem 4.

As indicated by Fig. 1, $r^{\#} \geqslant 1/(1-p)$ for $p \geqslant 0.694\ldots$. To prove $r^{\#} \geqslant 1/(1-p)$ for $p \geqslant 0.694\ldots$, and thus $p_c < 0.695$, we observe that $d(1/(1-p), p)$ is strictly decreasing for $p > 1 - 1/v$, where $v = -\frac{7}{3}\ln(14\ln 7/8/(7\ln 7/8 - 3)) - \frac{5}{9}$, i.e. for $p > 0.153\ldots$. Hence the claim follows from the fact $d(1/(1-p), p) = 0.99998\ldots$, for $p = 0.6945$.

5. Almost sure satisfiability

To prove almost sure satisfiability for $F_{2+p}(n, rn)$ we will first prove that, like random k-SAT, random $(2+p)$-SAT has a sharp threshold for all $p \in [0,1]$. Thus, analogously to Corollary 1, if $F_{2+p}(n, r_p^* n)$ is satisfiable with positive probability, then for $r < r_p^*$ it is satisfiable a.s. In order to determine such a value r_p^*, the key observation is that in analyzing a number of different algorithms on random 3-SAT instances [10, 11, 23, 1] one can view the clauses remaining after each step as the union of uniformly random sets of 1-clauses, 2-clauses and 3-clauses. Hence, one can readily analyze any one of these algorithms on random $(2+p)$-SAT since the input formula simply appears like an "intermediate" formula of a random 3-SAT execution. We will analyze the execution of the simplest such algorithm, called unit-clause (UC) [10], on random $(2+p)$-SAT. This will simplify the exposition greatly and, as we will argue in Section 6, this simplicity comes without a sacrifice: all the algorithms considered in [10, 11, 23, 1] yield the same lower bound for p_c, i.e. $p_c \geqslant 2/5$.

Let us start by establishing that $(2+p)$-SAT has a sharp threshold. Recall that $g_p(n, r)$ denotes the probability $F_{2+p}(n, rn)$ is satisfiable.

Theorem 2. *For every* $p \in [0,1]$, *there exists* $r_p(n)$ *such that for any* $\varepsilon > 0$,

$$\lim_{n\to\infty} g_p(n, r_p(n) - \varepsilon) = 1 \quad \textit{and} \quad \lim_{n\to\infty} g_p(n, r_p(n) + \varepsilon) = 0.$$

Proof. The proof uses the techniques developed in [22] to show a sharp threshold for random k-SAT, after addressing the following technical point.

For the machinery developed in [22] to work we need to be in a product probability space, i.e. the one where every clause appears in the formula independently of all other clauses. Since in the setting of random $(2+p)$-SAT there are $|B_3| \equiv N_3 = 8\binom{n}{3}$ potential 3-clauses, but only $|B_2| \equiv N_2 = 4\binom{n}{2}$ potential 2-clauses we will construct our random formula, $H_{2+p}(n, rn)$, by considering random trials over $\Omega(n)$ independent *copies* of each 2-clause for $p \in (0,1)$. Note that now rn will be the *expected* and not the exact number of clauses in the formula (we use this notation to facilitate comparison with $F_{2+p}(n, rn)$).

To form $H_{2+p}(n, rn)$ when $p = 0$ or 1, we simply include in the formula each clause of B_{2+p} independently of all others, with probability rn/N_{2+p} (and hence the expected number of clauses is rn). For $p \in (0,1)$ we proceed as follows. Let B_2^q denote the multiset containing q copies of each clause in B_2, where q is a given integer. For a given $p \in (0,1)$ let $B_{2+p} \equiv B_3 \cup B_2^{q(p)}$, where $q(p) = \lfloor (2(1-p)/3p)n \rfloor$, and note that B_{2+p} contains N_3/p (non-distinct) clauses. $H_{2+p}(n, rn)$ is formed by selecting each member of B_{2+p}, independently of all others, with probability $s = prn/N_3$. Hence, $H_{2+p}(n, rn)$ contains rn, not necessarily distinct, clauses on average. Having defined B_{2+p} in this manner, it will be rather straightforward to adapt the techniques in [22] to prove that if $h_p(n, r)$ is the probability that $H_{2+p}(n, rn)$ is satisfiable, then

Lemma 1. *For every* $p \in [0,1]$ *there exists a function* $r_p(n)$ *such that for any* $\varepsilon > 0$,

$$\lim_{n\to\infty} h_p(n, r_p(n) - \varepsilon) = 1 \quad \textit{and} \quad \lim_{n\to\infty} h_p(n, r_p(n) + \varepsilon) = 0.$$

Before proving Lemma 1 let us first show that indeed the sharp threshold for $h_p(n, r)$ implies a sharp threshold for $g_p(n, r)$. Recall that in $F_{2+p}(n, rn)$ there are prn 3-clauses and $(1-p)rn$ 2-clauses and let $\lambda^* = (1-p)rn/N_2$. Note now that for every $p \in (0,1)$ the following is true: for each clause in B_2, the probability that at least one of its $q(p)$ copies appears in $H_{2+p}(n, rn)$ is $\lambda = \lambda(p, r, n) = 1 - (1-s)^{q(p)}$. Hence, the number of *distinct* 2-clauses in $H_{2+p}(n, rn)$ is distributed as $\mathrm{Bin}(N_2, \lambda)$. Moreover, the fact

$$1 - \left(1 - \frac{a}{n^2}\right)^{bn} = \frac{ab}{n} + \mathrm{O}(n^{-2})$$

implies that $\lambda = \lambda^* + \mathrm{O}(n^{-2})$.

As a result, for $p \in [0,1)$ the number of distinct 2-clauses in $H_{2+p}(n, rn)$ is distributed as $\mathrm{Bin}(N_2, \lambda)$, where $N_2\lambda = (1-p)rn + \mathrm{O}(1)$. On the other hand, for $p \in (0,1]$ the number of 3-clauses appearing in $H_{2+p}(n, rn)$ is distributed as $\mathrm{Bin}(N_3, s)$, where $N_3 s = prn$. Therefore, by applying the Chernoff bound to the number of distinct 2-clauses and the number of 3-clauses, we see that for any $p \in [0,1]$ and any $r > \varepsilon > \varepsilon' > 0$, almost surely:

(i) In $H_{2+p}(n, (r-\varepsilon')n)$ the number of distinct 2-clauses is at least $(1-p)(r-\varepsilon)n$ and the number of 3-clauses is at least $p(r-\varepsilon)n$.

(ii) In $H_{2+p}(n,(r+\varepsilon')n)$ the number of distinct 2-clauses is at most $(1-p)(r+\varepsilon)n$ and the number of 3-clauses is at most $p(r+\varepsilon)n$.
Furthermore, it is clear that all formulae are equally likely, conditional on the number of distinct 2-clauses and the number of 3-clauses they contain. Therefore, as $g_p(n,m)$ is non-increasing in m and (i), (ii) hold for any $r>\varepsilon>\varepsilon'>0$, the sharp threshold for h_p is also a sharp threshold for g_p.

Finally, if $H_k(n,m)$ is defined analogously to $H_3(n,m)$, the number of k-clauses in $H_k(n,m)$ is distributed as $\mathrm{Bin}(N_k,m/N_k)$. Hence, exactly as in the above paragraph, a sharp threshold for $H_k(n,m)$ [22] yields one for $F_k(n,m)$ as was claimed in reporting Theorem 1.

As mentioned earlier, to prove the existence of a sharp threshold for random $(2+p)$-SAT we will use the techniques of [22]. In particular, in [22], Friedgut gives a characterization of properties of random k-SAT formulae that do not exhibit a sharp threshold and uses it to show that satisfiability is not such a property. The proof of the characterization is quite lengthy yet it is rather straightforward (but tedious) to check that a similar characterization can be derived for properties of random $(2+p)$-SAT formulae. Rather than taking this approach, we will instead use a general condition for a monotone property to have a coarse threshold, given by Bourgain in an appendix to [22], which can be readily used for our purposes.

To introduce Bourgain's Theorem, let us recall that a subset A of $\{0,1\}^N$ is called monotone if whenever $x\in A$, $x'\in\{0,1\}^N$, $x_i\leqslant x'_i$ for $i=1,\ldots,N$, then $x'\in A$. For $0\leqslant s\leqslant 1$, define μ_s to be the product measure on $\{0,1\}^N$ with weights $1-s$ at 0 and s at 1. Thus,

$$\mu_s(\{x\})=(1-s)^{N-j}s^j \quad \text{where } j=|\{i=1,\ldots,N\colon x_i=1\}|.$$

If A is monotone, then $\mu_s(A)$ is clearly an increasing function of s. (In our setting, A will be the property of unsatisfiability.)

Theorem 6 (Bourgain [22]). *Let $A\subset\{0,1\}^N$ be a monotone property and assume that*

$$\mu_s(A)=\tfrac{1}{2}, \tag{8}$$

$$s\frac{d\mu_s(A)}{ds}<C, \tag{9}$$

$$s=\mathrm{o}(1). \tag{10}$$

Then there is $\delta=\delta(C)$ such that either

$$\mu_s(x\in\{0,1\}^N\colon x \text{ contains } x'\in A \text{ of size } |x'|\leqslant 10C)>\delta \tag{11}$$

or there exists $x'\notin A$ of size $|x'|\leqslant 10C$ such that the conditional probability

$$\mu_s(x\in A\,|\,x\supset x')>\tfrac{1}{2}+\delta. \tag{12}$$

One can replace $\frac{1}{2}$ by any other $0<\alpha<1$ in (8); this results in 1/2 being replaced by α in (12).

Conditions (8) and (9) above imply that A has a coarse threshold (condition (10) is technical). On the other hand, (11) implies the existence of a "small and often-encountered witness" for membership in A, while (12) implies the existence of a "booster", a short substring that does not imply membership in A but which makes it substantially more likely. In our context, the witness would be a small satisfiable subformula, while the booster would be small satisfiable subformula with the property that, conditional on its presence, unsatisfiability is significantly more likely.

To prove Lemma 1 using Theorem 6 we first note that in our case $N = |B_{2+p}|$, A is the (monotone increasing) property of unsatisfiability and the generic x in Theorem 6 corresponds to a formula picked from the $H_{2+p}(n,m)$ model. Further note that if $\varepsilon < \mu_A(s) < 1-\varepsilon$, i.e. if the probability of satisfiability is constant, then $s = \mathrm{o}(1)$ and in particular the expected number of clauses in the formula $m = rn + \mathrm{o}(n)$. To see this note that for any $p \in [0,1]$, there exists r^* such that if $m = r^* n$ then the expected number of satisfying truth assignments is o(1). In the opposite direction, note that for any $p \in [0,1]$ if $r < 1$ then by removing a random literal from each 3-clause we are a.s. left with a random 2-SAT formula with $m = rn$ clauses, i.e. an a.s. satisfiable formula.

Therefore, it suffices to prove that if s is such that $m = \Theta(n)$ then neither of (11), (12) can occur. Excluding (11) is straightforward as it suffices to observe that any unsatisfiable formula on q variables must contain $q+1$ distinct clauses. A straightforward calculation now implies that for any constant q the expected number of such subformulae in $H_{2+p}(n,rn)$ is o(1).

Excluding the possibility of (12) is non-trivial and amounts to showing that no satisfiable subformula of constant size can have a significant "influence" on the satisfiability of a random $H_{2+p}(n,rn)$ formula. The steps in this last proof are identical to those in the proof of Corollary 5.3 of Friedgut [22] and we only outline them below in order to avoid redundancy.

Let the size of the purported x' in (12) be $|x'| = q$. To form the conditional probability space considered in (12) we will form a $H_{2+p}(n,rn)$ formula by first fixing a copy of the claimed subformula x' on variables $V_x = \{v_1, v_2, \ldots, v_q\}$ and then adding every other clause in B_{2+p} with probability s. We want to show that for any fixed $\delta > 0$, the probability that the resulting formula is unsatisfiable is smaller than $1/2 + \delta$. There are two key things we need to observe. The first one is that, by monotonicity, the probability that the subformula on variables $V - V_x$ is unsatisfiable is at most $1/2$. The second is that, as an easy calculation can show, there exists a constant M such that with probability at least $1 - \delta/2$, at most M of the random clauses in the formula contain a variable from V_x and, further, every such clause contains at most one variable from V_x. As a result, with probability $1 - \delta/2$ conditioning on the occurrence of x' in the formula does not increase the probability of unsatisfiability more than adding M random clauses of length 1.

To conclude the proof it suffices to show that adding a constant number of 1-clauses to the formula cannot increase the probability of satisfiability by more than o(1). This is done in two steps. The first step amounts to showing that for any monotone property, if $s' = s + \mathrm{o}(\sqrt{sN})$ then $\mu_{s'}(A) - \mu_s(A) = \mathrm{o}(1)$. In our case, such an increase in s would

correspond to an (expected) addition of $o(\sqrt{n})$ clauses selected randomly from B_{2+p}. The second step amounts to showing that adding a constant number M of 1-clauses to a random $H_{2+p}(n,rn)$ formula does not increase the probability of unsatisfiability more than adding an exponential, in M, number of clauses selected randomly from B_{2+p}. In particular, these two last steps are lemmata 5.6 and 5.7, respectively, in [22]. □

5.1. *The* Unit Clause *algorithm*

The Unit Clause (uc) algorithm, presented below, was introduced and analyzed for random 3-SAT by Chao and Franco [10]. The algorithm makes n iterations, in each one permanently setting one variable. In the following, "at time t" means after t such iterations have been performed (i.e. after t variables have been set) and step t will refer to the step performed between time t and $t+1$. As soon as a clause is satisfied we consider it removed from the formula, never to be considered again. On the other hand, as soon as a literal in an i-clause c becomes dissatisfied, we consider that literal removed from c and we consider c an $(i-1)$-clause. We let $\mathscr{C}_i(t)$ denote the set of i-clauses remaining at time t.

Unit Clause

- $V \leftarrow \{x_1,\ldots,x_n\}$.
- For $t=0,\ldots,n-1$
  1. If $\mathscr{C}_1(t)\neq\emptyset$ then choose randomly a literal $l\in\mathscr{C}_1(t)$
     else choose randomly a variable $v\in V$ and
     take l to be $v,\bar{v}$ with equal probability.
  2. Set var(l) so as to satisfy l.
  3. $V \leftarrow V - \mathrm{var}(l)$.
  4. Remove all clauses in which l appears. (They are satisfied)
  5. Remove $\bar{l}$ from all clauses. ("Shrunk" clauses "move" from $\mathscr{C}_i(t)$ to $\mathscr{C}_{i-1}(t+1)$)

At each substep of type 5, uc might generate a clause of length 0 (contradiction) and clearly such a clause will never be removed. On the other hand, if this never happens then uc finds a satisfying truth assignment, in which case we say that it *succeeds*.

Let $V(t)$ denote the set of variables not assigned a truth value at time t and let $C_i(t)=|\mathscr{C}_i(t)|$. Recall that for a set of Boolean variables V, $B_i(V)$ denotes the set of all non-trivial i-clauses on the variables of V. Also, let $L(V)$ denote the set of $2|V|$ literals on the variables of V.

In analyzing uc it will be convenient to view a random $(2+p)$-SAT formula with m clauses, as constructed through the following sequence of random choices: for each $1 \leqslant j \leqslant (1-p)m$, two distinct, non-complementary literals are picked uniformly at random to form the jth 2-clause; for each $1 \leqslant j \leqslant pm$, three distinct, non-complementary literals are picked uniformly at random to form the jth 3-clause. As we proceed through the execution of the algorithm, in each step, we will only expose those properties of the random choices made in forming the formula that are necessary

to carry out the current step. Specifically, in order to carry out steps 4 and 5, for each i-clause we first expose the answer to the following question: "Does this clause contain one of $l, \bar{l}$?" If the answer is "No", we consider this random clause to be distributed according to the appropriate conditional probability distribution, i.e. we consider it to be a uniformly random set of i distinct, non-complementary literals from $L(V(t)-\mathrm{var}(l))$. If the answer is "Yes", we proceed to ask if the clause contains l. If the answer is "Yes" we delete this clause and it no longer concerns us. If the answer is "No" we remove one literal from this clause and consider the remainder as a uniformly random set of $i-1$ distinct, non-complementary literals from $L(V(t)-\mathrm{var}(l))$. Thus, we have

Lemma 2. *For every $0 \leqslant t \leqslant n$, conditional on all information exposed thus far, each clause in $\mathscr{C}_i(t)$ contains a uniformly random set of i distinct, non-complementary literals from $L(V(t))$.*

We note that (an analogue of) Lemma 2 holds for all algorithms in [10, 11, 23, 1] when applied to $F_{2+p}(n,m)$.

Lemma 3, below, follows by combining Theorems 4 and 6 in [10]. Roughly speaking, it asserts that as long as the density of the 2-SAT subformula formed by the clauses in $\mathscr{C}_2$ stays below 1, UC has positive probability of never generating an empty clause (and, at the same time, the probability of there being no 1-clause present at the beginning of a given step is also positive).

Lemma 3. *Fix $\delta, \varepsilon > 0$ and let $t_0 = n - \lfloor \varepsilon n \rfloor$. The probability that for all $0 \leqslant t \leqslant t_0$, $C_2(t) < (1-\delta)(n-t)$ and $\mathscr{C}_0(t_0) \cup \mathscr{C}_1(t_0) \neq \emptyset$, is at most $1-\rho$ for some $\rho = \rho(\delta, \varepsilon) > 0$.*

Thus, we see that in order to bound the probability of UC failing in the first $t_0 = n - \lfloor \varepsilon n \rfloor$ steps it suffices to trace the evolution of $C_2(t)$ and determine for which values of p, r it stays uniformly away from $(n-t)$ for all $0 \leqslant t \leqslant t_0$. To deal with the last $\lfloor \varepsilon n \rfloor$ steps we will show that a.s. at time t_0 there are so few 2- and 3-clauses remaining, that we are left with an "easy to satisfy" formula.

Let $\vec{U}(t) = \langle C_2(t), C_3(t) \rangle$ be a vector describing the number of 2- and 3-clauses at time t and let $\mathbf{H}(t) = \langle \vec{U}(0), \ldots, \vec{U}(t) \rangle$ be a $2 \times (t+1)$ matrix describing the entire history of the number of 2-clauses and 3-clauses up to time t. For random variables X, Y let us write $X \stackrel{D}{=} Y$ if for every value x in the domain of X, $\Pr[X = x] = \Pr[Y = x]$. Finally, let $\mathrm{Bin}(N, s)$ denote the binomial random variable with N trials each having probability of success s.

Lemma 4. *Let $\Delta C_i(t) = C_i(t+1) - C_i(t)$. For all $0 \leqslant t \leqslant n-3$, conditional on $\mathbf{H}(t)$,*

$$\Delta C_3(t) = -X, \tag{13}$$

$$\Delta C_2(t) = Y - Z, \tag{14}$$

where

$$X \stackrel{D}{=} \mathrm{Bin}\left(C_3(t), \frac{3}{n-t}\right), \quad Y \stackrel{D}{=} \mathrm{Bin}\left(C_3(t), \frac{3}{2(n-t)}\right),$$

$$Z \stackrel{D}{=} \mathrm{Bin}\left(C_2(t), \frac{2}{n-t}\right). \tag{15}$$

Proof. Intuitively, each negative term in (13), (14) represents the number of clauses leaving $\mathscr{C}_i(t)$ during step t, either as satisfied or as shrunk, while the positive term expresses the fact that the clauses leaving $\mathscr{C}_3(t)$, with probability $1/2$ move to $\mathscr{C}_2(t+1)$. To prove the lemma we first claim that for every $0 \leqslant t \leqslant n-1$ the literal satisfied during step t is chosen uniformly at random among the literals in $L(V(t))$. To prove this, we simply note that when $\mathscr{C}_1(t) \neq \emptyset$ the claim follows by Lemma 2 applied to $\mathscr{C}_1(t)$, while when $\mathscr{C}_1(t) = \emptyset$ it follows from the definition of the algorithm.

Let l be the literal chosen to be satisfied during step t. As l is uniformly random among the literals in $L(V(t))$, by Lemma 2, we know that every clause $c \in \mathscr{C}_i(t)$, $i = 2$, 3, contains one of $l, \bar{l}$, independently of all other clauses and with the same probability. As there are $n-t$ unset variables, if c has i literals then this probability is $i/(n-t)$. This yields the negative terms in (13), (14) as each clause containing one of $l, \bar{l}$ is removed from the set it belonged at time t. To get the positive term we note that as l is uniformly random, by Lemma 2, if $c \in \mathscr{C}_3(t)$ contains one of $l, \bar{l}$ then it contains $\bar{l}$ with probability $1/2$. □

Lemma 4 defines a "mean path" for the sequence of random variables $C_i(0), C_i(1), \ldots, i = 2, 3$. Moreover, from (15), we see that (i) $\mathbf{E}[\Delta C_i(t)] = \mathrm{O}(1)$, (ii) $\Delta C_i(t)$ is concentrated around its expectation, and (iii) $\mathbf{E}[\Delta C_i(t)]$ is a "smooth" function of $t, C_2(t), C_3(t)$. These facts will allow us to approximate the evolution of each sequence C_i by a system of differential equations. In particular, let functions $c_2(x), c_3(x)$ satisfy

$$c_3'(x) = -\frac{3}{1-x} c_3(x), \quad c_3(0) = rp,$$

$$c_2'(x) = \frac{3}{2(1-x)} c_3(x) - \frac{2}{1-x} c_2(x), \quad c_2(0) = r(1-p).$$

Solving the two differential equations implies

$$c_3(x) = rp(1-x)^3, \tag{16}$$

$$c_2(x) = \tfrac{1}{2} r(3px - 2p + 2)(1-x)^2. \tag{17}$$

Lemma 5, below, asserts that c_2, c_3 approximate $C_2(t), C_3(t)$ within $\mathrm{o}(n)$. An analogous, but less precise, statement is given without proof for 3-SAT (i.e. $p = 1$) in [10]. We will prove Lemma 5 by applying a Theorem of Wormald [37] (stated as Theorem A.1 in the appendix for completeness).

Lemma 5. *For any* $\varepsilon>0$, *if* UC *is applied to* $F_{2+p}(n,rn)$ *then a.s. for* $0 \leqslant t \leqslant t-\lfloor \varepsilon n \rfloor$,

$$C_i(t) = c_i(t/n)\cdot n + o(n) \quad \text{for } i = 2,3. \tag{18}$$

Proof. By Lemma 4, we can apply the Chernoff bound to bound $\Pr[\Delta C_i(t) > n^{1/5} | \mathbf{H}(t)]$, for each $i=2,3$. Thus, for any $\varepsilon>0$ the lemma follows by applying Theorem 7 with $k=2$, $Y_i(t)=C_{i+1}(t)$, $C=r$, $m=n-3$, $f_1(s,z_1,z_2)=(3z_2/2(1-s))-(2z_2/(1-s))$, $f_2(s,z_1,z_2)=-(3z_2/2(1-s))$ and D defined by $-\varepsilon<s<1$, $-\varepsilon<z_i<r$, for $i=1,2$. □

We can now determine values of r, p for which $F_{2+p}(n,rn)$ is satisfiable with positive probability. In particular, we claim that this holds for all r, p such that for all $x\in[0,1]$,

$$\tfrac{1}{2}r(3px-2p+2)(1-x)<1. \tag{19}$$

Postponing the proof of this claim for a moment let us see what lower bounds it implies for r_p.

- For $p \leqslant 2/5$, the left-hand side of (19) is non-increasing with x. Hence, (19) holds iff it holds for $x=0$, i.e. iff

$$r < \frac{1}{1-p}.$$

  Thus, by Theorem 2, $r_p \geqslant 1/(1-p)$ for $p \leqslant 2/5$. Since $r_p \leqslant 1/(1-p)$ for all $p\in[0,1)$, we get $r_p = 1/(1-p)$ for $p \leqslant 2/5$, i.e. Theorem 3.
- For $p>2/5$, the left-hand side of (19) is unimodal for $x\in[0,1]$, the unique maximum occurring at $x=(5p-2)/6p$. For such x, (19) holds iff

$$r < \frac{24p}{(p+2)^2}.$$

Thus, by Theorem 2, we have $r_p \geqslant 24p/(p+2)^2$ for $p>2/5$, i.e. the lower bound in Theorem 4.

We now need to prove our earlier claim regarding the role of (19). For a given $\varepsilon>0$, let $t_0 = n-\lfloor \varepsilon n \rfloor$. Now, assume that r, p are such that for some $\delta>0$ and all $x\in[0,1]$, $\frac{1}{2}r(3px-2p+2)(1-x) \leqslant 1-\delta$. Note that if this is true then $r<8/3$. Lemma 5 then implies that (i) a.s. for all $0 \leqslant t \leqslant t_0$, $C_2(t)<(1-\delta)(n-t)$ and that (ii) a.s. $C_2(t_0)+C_3(t_0)<3\varepsilon^2 rn$. The first fact along with Lemma 3 imply that with probability $\rho=\rho(\delta)-o(1)$, $\mathscr{C}_0(t_0)\cup\mathscr{C}_1(t_0)=\emptyset$. The second fact along with Lemma 2 imply that if at time t_0 we stop the execution of UC and just delete one literal at random from each clause in $\mathscr{C}_3(t_0)$, the resulting 2-clauses along with those in $\mathscr{C}_2(t_0)$ form $F_2(n-t_0,m)$, where a.s. $m<3\varepsilon^2 rn$.

Thus, in conclusion, we get that for each fixed $\varepsilon>0$, if we stop the execution of UC on $F_{2+p}(n,rn)$ at time t_0 and delete one literal at random from each remaining 3-clause, with positive probability we will be left with a random 2-SAT formula with $\lfloor \varepsilon n \rfloor = \Omega(n)$ variables and fewer than $3\varepsilon^2 rn$ clauses. For $\varepsilon>0$ sufficiently small, since r is bounded, this formula will a.s. be satisfiable, completing the proof of the claim.

6. Discussion

6.1. Why 2/5?

Let us consider an execution of UC on $F_2(n, rn)$. Since each variable originally appears on average in $2r$ clauses, after the first θn steps of the execution, $\theta \to 0$, we will have roughly $rn - 2r\theta n$ 2-clauses remaining, over $n - \theta n$ variables, i.e. a 2-SAT formula with density $r(1-2\theta)/(1-\theta) = r' < r$. Since this holds for any $r > 0$, by applying this argument "inductively", we see that the density of the 2-SAT formula formed by the remaining clauses drops throughout the execution of UC.

Note now, that the condition in Lemma 3 states that as long as the density of this subformula stays below 1, there is positive probability that UC generates no contradictions in processing it. Hence, the fact that the 2-SAT formula becomes sparser during the execution of UC suggests that there is room to make the algorithm work harder. That is, the algorithm would still have positive probability of success if the density of the underlying 2-SAT subformula simply remained below 1, instead of actually dropping during the execution. This is where the 3-clauses come in.

If $\alpha < 1$ and we take a conjunction of $F_2(n, \alpha n)$ with $F_3(n, \lambda\alpha n)$ and apply UC to it, then in the first θn steps of the execution we have: roughly $2\alpha\theta n$ of the original 2-clauses are removed (satisfied or shrunk), while roughly $\frac{3}{2}\lambda\alpha\theta n$ (shrunk) 3-clauses end up in the 2-SAT subformula. Thus, if $\lambda \leqslant 2/3$ the 2-SAT formula will have no more than $\alpha(1-\theta)n$ 2-clauses over $(1-\theta)n$ variables, i.e. its density will remain at most α. Again, this argument can be applied "inductively" (yet now things are already a bit better as slightly fewer 3-clauses will shrink to 2-clauses). Thus, we see that up to $\frac{2}{3}\alpha n$ 3-clauses can be essentially be "piggybacked" on $F_2(n, \alpha n)$. In $(2+p)$-SAT terms, this corresponds to $r(1-p) = \alpha$ and $rp = \frac{2}{3}\alpha$, i.e. $p = 2/5$.

This argument also suggests that the "hardest" part of the execution for UC in dealing with such a formula is the very beginning. At that point, the rate at which 1-clauses are generated can be arbitrarily close to 1 as $\alpha \to 1$. This last fact also suggests that any algorithm which maintains "uniform randomness" in the sense of Lemma 2 and takes care of 1-clauses first cannot give an improved lower bound for p_c (this can be made rigorous). The intuition is that, in the beginning of the execution, every time such an algorithm sets the value of some variable "freely", it can expect $1/(1-\alpha)$ forced steps to follow; even if somehow no 3-clauses at all became 2-clauses in the free step, in all the following steps there is nothing that can be done about 3-clauses becoming 2-clauses. Thus, in some sense, as $\alpha \to 1$ all such algorithms tend to UC. As a result, if we have $(1-\varepsilon)n$ random 2-clauses and λn random 3-clauses, where $\lambda > 2/3$, then such an algorithm fails a.s. for some $\varepsilon = \varepsilon(\lambda) > 0$. In particular, this is the case for all the algorithms in [10, 11, 23, 1].

Since the appearance of an extended abstract of this article [2], our results have provided some feedback to the statistical mechanics community [33, 6, 35]. In fact, Biroli et al. [6], using a variational argument in combination with the replica method, showed $p_c = 2/5$ contrary to the results in [34]. Again, this is not a rigorous result but the authors claim that in some sense it is "more rigorous" than the results in [34].

We feel that determining p_c is a very interesting problem. If $p_c > 2/5$, this would shed much needed light in the applications of the replica method on random $(2+p)$-SAT and random k-SAT. If $p_c = 2/5$, it would be very insightful if we can draw analogies between the "combinatorial" arguments leading to this fact and the "statistical mechanics" ones.

Acknowledgements

This work was supported in part by the National Science and Engineering Research Council of Canada and by the EU ESPRIT Long-term Research Project ALCOM-IT (Project no. 20244). We would like to thank an anonymous referee for his suggestion to use Bourgain's theorem to prove our Theorem 2. Dimitris Achlioptas would like to thank Michael Molloy for numerous helpful discussions.

Appendix

In the statement of Theorem 7, below, asymptotics denoted by o and O are for $n \to \infty$ but uniform over all other variables. In particular, "uniformly" refers to the convergence implicit in the o() terms. For a random variable X, we say $X = \mathrm{o}(f(n))$ *always* if $\max\{x \mid \Pr[X = x] \neq 0\} = \mathrm{o}(f(n))$. We say that a function f satisfies a *Lipschitz condition* on $D \subseteq \mathbb{R}^j$ if there exists a constant $L > 0$ such that $|f(u_1, \ldots, u_j) - f(v_1, \ldots, v_j)| \leqslant L \sum_{i=1}^{j} |u_i - v_i|$, for all $(u_1, \ldots, u_j)$ and $(v_1, \ldots, v_j)$ in D.

Theorem A.1 (Wormald [37]). *Let $Y_i(t)$ be a sequence of real-valued random variables, $1 \leqslant i \leqslant k$ for some fixed k, such that for all i, all t and all n, $|Y_i(t)| \leqslant Cn$ for some constant C. Let $\mathbf{H}(t)$ be the history of the sequence, i.e. the matrix $\langle \vec{Y}(0), \ldots, \vec{Y}(t) \rangle$, where $\vec{Y}(t) = (Y_1(t), \ldots, Y_k(t))$.*

Let $I = \{(y_1, \ldots, y_k)\colon \Pr[\vec{Y}(0) = (y_1 n, \ldots, y_k n)] \neq 0$ for some $n\}$. Let D be some bounded connected open set containing the intersection of $\{(s, y_1, \ldots, y_k)\colon s \geqslant 0\}$ with a neighborhood of $\{(0, y_1, \ldots, y_k)\colon (y_1, \ldots, y_k) \in I\}$. [4]

Let $f_i\colon \mathbb{R}^{k+1} \to \mathbb{R}$, $1 \leqslant i \leqslant k$, and suppose that for some $m = m(n)$,

(i) *for all i and uniformly over all $t < m$,*

$$\mathbf{E}(Y_i(t+1) - Y_i(t) | \mathbf{H}(t)) = f_i(t/n, Y_0(t)/n, \ldots, Y_k(t)/n) + \mathrm{o}(1) \quad always;$$

(ii) *for all i and uniformly over all $t < m$,*

$$\Pr[|Y_i(t+1) - Y_i(t)| > n^{1/5} | \mathbf{H}(t)] = \mathrm{o}(n^{-3}) \quad always;$$

[4] That is, after taking a ball around the set I, we require D to contain the part of the ball in the halfspace corresponding to $s = t/n \geqslant 0$.

(iii) *for each i, the function f_i is continuous and satisfies a Lipschitz condition on D. Then,*

(a) *for $(0,\hat{z}^{(0)},\ldots,\hat{z}^{(k)})\in D$ the system of differential equations*

$$\frac{\mathrm{d}z_i}{\mathrm{d}s} = f_i(s,z_0,\ldots,z_k), \quad 1 \leqslant i \leqslant k$$

has a unique solution in D for $z_i:\mathbb{R}\to\mathbb{R}$ passing through $z_i(0)=\hat{z}^{(i)}$, $1 \leqslant i \leqslant k$, and which extends to points arbitrarily close to the boundary of D;

(b) *almost surely*

$$Y_i(t) = z_i(t/n)\cdot n + \mathrm{o}(n),$$

uniformly for $0 \leqslant t \leqslant \min\{\sigma n, m\}$ and for each i, where $z_i(s)$ is the solution in (a) with $\hat{z}^{(i)} = Y_i(0)/n$, and $\sigma = \sigma(n)$ is the supremum of those s to which the solution can be extended.

Note: The theorem remains valid if the reference to "always" in (i),(ii) is replaced by the restriction to the event $(t/n, Y_0(t)/n,\ldots,Y_k(t)/n)\in D$.

References

[1] D. Achlioptas, Setting two variables at a time yields a new lower bound for random 3-SAT, in: 32nd Ann. ACM Symp. on Theory of Computing, Portland, OR, 2000, ACM, New York, 2000, pp. 28–37.

[2] D. Achlioptas, L.M. Kirousis, E. Kranakis, D. Krizanc, Rigorous results for random $(2+p)$-SAT, in: RALCOM '97, Santorini, 1997, 1997, pp. 1–13.

[3] D. Achlioptas, L.M. Kirousis, E. Kranakis, D. Krizanc, M. Molloy, Y. Stamatiou, Random constraint satisfaction: a more accurate picture, Constraints, to appear.

[4] D. Achlioptas, M. Molloy, The analysis of a list-coloring algorithm on a random graph, in: 38th Ann. Symp. on Foundations of Computer Science, Miami, FL, 1997, IEEE Comput. Soc. Press, Los Alamitos, CA, 1997, pp. 204–212.

[5] P. Beame, T. Pitassi, Propositional proof complexity: past, present, and future, Bull. Eur. Assoc. Theoret. Comput. Sci. EATCS 65 (1998) 66–89.

[6] G. Biroli, R. Monasson, M. Weigt, A variational description of the ground state structure in random satisfiability problems, Eur. Phys. J. B 14 (2000) 551–568.

[7] B. Bollobás, C. Borgs, J. Chayes, Jeong Han Kim, D.B. Wilson, The scaling window of the 2-SAT transition, 1999, manuscript.

[8] A.Z. Broder, A.M. Frieze, E. Upfal, On the satisfiability and maximum satisfiability of random 3-CNF formulas, in: 4th Ann. ACM-SIAM Symp. on Discrete Algorithms, Austin, TX, 1993, ACM, New York, 1993, pp. 322–330.

[9] M.-Te. Chao, J. Franco, Probabilistic analysis of two heuristics for the 3-satisfiability problem, SIAM J. Comput. 15 (4) (1986) 1106–1118.

[10] M.-Te. Chao, J. Franco, Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k-satisfiability problem, Inform. Sci. 51 (3) (1990) 289–314.

[11] V. Chvátal, B. Reed, Mick gets some (the odds are on his side), in: 33th Ann. Symp. on Foundations of Computer Science, Pittsburgh, PA, 1992, IEEE Comput. Soc. Press, Los Alamitos, CA, 1992, pp. 620–627.

[12] S.A. Cook, The complexity of theorem-proving procedures, in: 3rd Ann. ACM Symp. on Theory of Computing, Shaker Heights, OH, 1971, ACM, New York, 1971, pp. 151–158.

[13] S.A. Cook, D.G. Mitchell, Finding hard instances of the satisfiability problem: a survey, in: Satisfiability Problem: Theory and Applications (Piscataway, NJ, 1996), Amer. Math. Soc., Providence, RI, 1997, pp. 1–17.

[14] M. Davis, G. Logemann, D. Loveland, A machine program for theorem-proving, Comm. ACM 5 (1962) 394–397.
[15] M. Davis, H. Putnam, A computing procedure for quantification theory, J. Assoc. Comput. Mach. 7 (1960) 201–215.
[16] O. Dubois, Y. Boufkhad, A general upper bound for the satisfiability threshold of random r-SAT formulae, J. Algorithms 24 (2) (1997) 395–420.
[17] O. Dubois, Y. Boufkhad, J. Mandler, Typical random 3-SAT formulae and the satisfiability threshold, in: 11th Ann. ACM-SIAM Symp. on Discrete Algorithms, San Francisco, CA, 2000, ACM, New York, 2000, pp. 126–127.
[18] A. El Maftouhi, W. Fernandez de la Vega, On random 3-sat, Combin. Probab. Comput. 4 (3) (1995) 189–195.
[19] W. Fernandez de la Vega, On random 2-SAT, 1992, manuscript.
[20] J. Franco, Probabilistic analysis of the pure literal heuristic for the satisfiability problem, Ann. Oper. Res. 1 (1984) 273–289.
[21] J. Franco, M. Paull, Probabilistic analysis of the Davis–Putnam procedure for solving the satisfiability problem, Discrete Appl. Math. 5 (1) (1983) 77–87.
[22] E. Friedgut, Sharp thresholds of graph properties, and the k-SAT problem, J. Amer. Math. Soc. 12 (1999) 1017–1054.
[23] A.M. Frieze, S. Suen, Analysis of two simple heuristics on a random instance of k-SAT, J. Algorithms 20 (2) (1996) 312–355.
[24] A. Goerdt, A threshold for unsatisfiability, J. Comput. System Sci. 53 (3) (1996) 469–486.
[25] A. Goldberg, On the complexity of the satisfiability problem, in: 4th Workshop on Automated Deduction, Austin, TX, 1979, 1979, pp. 1–6.
[26] S. Janson, Y.C. Stamatiou, M. Vamvakari, Bounding the unsatisfiability threshold of random 3-SAT, Random Structures Algorithms 17 (2) (2000) 103–116.
[27] A. Kamath, R. Motwani, K. Palem, P. Spirakis, Tail bounds for occupancy and the satisfiability threshold conjecture, Random Structures Algorithms 7 (1) (1995) 59–80.
[28] A.C. Kaporis, L.M. Kirousis, Y. Stamatiou, M. Vamvakari, M. Zito, The unsatisfiability threshold revisited, submitted.
[29] L.M. Kirousis, E. Kranakis, D. Krizanc, Y. Stamatiou, Approximating the unsatisfiability threshold of random formulas, Random Structures Algorithms 12 (3) (1998) 253–269.
[30] C.J.H. McDiarmid, On a correlation inequality of Farr, Combin. Probab. Comput. 1 (1992) 157–160.
[31] R. Monasson, R. Zecchina, Entropy of the K-satisfiability problem, Phys. Rev. Lett. 76 (21) (1996) 3881–3885.
[32] R. Monasson, R. Zecchina, Statistical mechanics of the random K-satisfiability model, Phys. Rev. E 56 (2) (1997) 1357–1370.
[33] R. Monasson, R. Zecchina, Tricritical points in random combinatorics: the $(2+p)$-SAT case, J. Phys. A: Math. and Gen. 31 (46) (1998) 9209–9217.
[34] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Phase transition and search cost in the $(2+p)$-SAT problem, 4th Workshop on Physics and Computation, Boston, MA, 1996.
[35] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, Determining computational complexity from characteristic "phase transitions", Nature 400 (6740) (1999) 133–137.
[36] B. Selman, D.G. Mitchell, H.J. Levesque, Generating hard satisfiability problems, Artificial Intelligence 81 (1–2) (1996) 17–29.
[37] N.C. Wormald, Differential equations for random processes and random graphs, Ann. Appl. Probab. 5 (4) (1995) 1217–1235.
[38] M. Zito, Randomised techniques in combinatorial algorithmics, Ph.D. Thesis, Department of Computer Science, University of Warwick, 1999.